APPLICATION OF COMPONENT PROGRAMMING IN A NETWORKING ENVIRONMENT, FOR THE ADVANCEMENT OF HIGHER EDUCATION*

Benny Hardjono, B.EE., M.Eng**

Abstract

To spread the application of component programming for the advancement of higher education, a basic theory of component programming (e.g. with MS Visual Basic) as well as three practical examples, and their design considerations, are presented in this paper. These include: library automation (with facilities such as: bar-coding, catalogue and borrowed-book search, circulation management, easily tailored report etc.), automated quiz, and two examples of simulation program – all with graphical interface and currently are running in a networking environment (using MS Windows NT 4, as server).

1. Introduction

There are many ready-made software packages available in the market nowadays, despite this, few are immediately ready to be utilized, in an Asian context (other than word processors and spread sheet software), to increase academic excellence. Due to reasons, just to name a few, such as: low local currency- to -US dollar exchange rate / high implementation cost [ref. table 1, shows a number of software price and licensing cost for single and multi user, as an example – to highlight the problem], inflexibility for adaptation and development, low adaptability for a local networking environment, under developed human resources, e.g. difficulty in reading English-manuals and the lack of urgency in buying original softwares.

[1]The arrival of component programming, in the 1990's, could be one of the solutions to overcome these problems, as it provides a relatively easy way to develop custom-made software for local use. However, this information might not have been well publicized or those who had heard of it were not convinced that it could be a solution. Further more, not many computer programmers in the higher education institutions are familiar or interested enough to develop their own software – since many pirated software (for single user on single machine) or networkable (for multi users) software are available in their country (especially in Asia). It is a common knowledge, that the usage of pirated software is against the international law, and that is usually plagued with various problems and limitations, such as the lack of manuals and inflexibility for further development. However, many people still violate the law because they could only see the short-term advantage and often fail to grasp, and that by making their own customized software is a long-term solution, which avoids piracy and overcomes the need for licensing cost. In short, making their own software is, at least, cheaper and relatively easier for further development.

---

* This paper is an adapted version (originally was submitted and accepted for the Seminar titled "Technology and Higher Education", hosted by Philippine Christian University & ACUCA - Association of Christian University & Colleges in Asia, May 12-13, 1999, at Hidden Valley Resort, Manila.)

** Dosen Tetap Jurusan Teknik Informatika dan Kepala Lab. Komputer UPH

[1] from Sidola, Bandung, Indonesia (1999) and www.r-tech.com, Oct 18 2002

Table 1. Example of academic pricing for a number of software packages

Software Name	**Academic Price (single point) in US $**	**Networkable (multi user) e.g. Windows NT, with licensing cost**
Office XP (SE) Standard	149	No
Office XP Prof.	214.95	No
Office XP Standard	174.95	No
Visual Basic.Net 2002 Standard	74.95	No
Visual Basic Enterprise V6.0 with plus Pack	184.95	No
Access 97[1]	25 (per point)	Open B (150-499)
Protel[1]	5371	Single point
Protel	4995	Single point
OrCAD[1] (complete)	2162	Up to 4 points
Winnebago Spectrum (Library Management Sofware)	5000 (only OPAC & Catalog modules – not including Circulation ect.)	Installed on one server only (database- desk top type i.e. Foxpro)
SQL Server v.7	4239	25 points
SQL Server v.7	729	5 points
Windows NT	419	5 points
Windows NT	589	10 points

In order to spread the use of component programming, a basic theory of component programming (e.g. with MS Visual Basic) as well of three practical examples are presented in this paper. These include: library automation (OMEGA Library Management System, with facilities such as: bar-coding, catalogue and borrowed-book search, circulation management, easily tailored reports etc.), automated quiz (AutoQuiz), with security aspects considered [Firdaus and Tjong, 1997] and two examples of simulation program – all with graphical interface and are currently running (year 2000 compatible) in a networking environment (using MS Windows NT 4 server) at Maranatha Christian University (MCU) and at Universitas Pelita Harapan (UPH). The examples presented here are by far perfect, and have not used all the advanced capabilities of component programming [Hardjono, 1995, 1997-1998], they are merely examples to attract more computer programmers to make their own custom-made software and to gather moral and financial support from the institution's executives and administrators – so that all can work together for the betterment of higher education.

2. Basic Theory

A component is a physical encapsulation of one or more services that are made available through its interfaces. It can be an *.exe file or *.dll file, a collection of

database triggers stored procedures, or any number of other physical software entities. A component is defined by the services it provides and how it interacts with other components. The internal construction and implementation are hidden from the outside world and, in this sense, components share the modular and encapsulating characteristics of business Objects [Microsoft, *Visual Basic Guide to Building Client/Server*, 1997].

Business Objects can be thought as specialized application Objects. They are self-contained, and in the services model of a three-tier architecture, they map to the second tier, the tier that houses the organization's business rules [Goren C., 1996]. Visual Basic (available in the market since 1991, for MS Windows, Perry, 1994) is one of the component programming software, which utilizes Objects in form of Control, Form, and Database, mentioned before. As a result it is also categorized as Object Oriented Programming (more commonly, abbreviated as OOP), because in OOP, an Object is created first along with its Property and the programming Code is written last. This is different than the conventional approach (such as, Turbo Pascal, or Fortran 77). It is encapsulated because it can contain both Data and programming Code (contained in e.g. Module and Form), hence it tends to have easier control than the past – traditional programming method [Microsoft, 1997]. Other characteristics embedded in Objects, are Event, Property and Method. Event is all the actions (noun), which an Object can do (e.g. a click of a button on Form). Next, Property, it is a character which can explain the attribute(s) in an Object. While, the third, Method, is all the verbs which can be commanded / carried out by the Object. A group of programming codes can be executed while certain events are happening. An Object is an instance of a Class. A Class will define the interfacing of an Object, whether it is Public (for general usage) and the kinds of condition this Object can be made. The details of a Class can be seen through an Object Browser (part of Visual Basic programming Menu), which are stored in type libraries. Reference is needed to use an Object, this is called Object Variable. Object variable's execution speed is depended on its Binding. There are three types of Binding, for example an Object declared as Dim var_name As classname uses the fastest binding. It is recommended to use late Binding for calls to the Internet Explorer Scripting Object Model. Using early Binding (fastest) will almost certainly cause compatibility problems in the future, while late Binding will always work [Microsoft, 1997]. In other containers, it is possible to make use of early Binding. Accessing properties and methods of a control is faster, if a variable declared with the same type as the control (for example, As TreeView or As CommandButton) is used, this is because Visual Basic can use early Binding. But Visual Basic must use late Binding to access properties and methods of a control assigned to a variable declared As Control. Due to this reason, a program must be planned well before the actual programming code is written.

Form and Control are two examples of Object, which have the above three characteristics. Even though Control is an independent Object, there is a parent and child relationship between Form and Control (A Form can contain Control). An additional feature in OOP is Polymorphism. With this feature, a programmer, without having to know which Class an Object owned can manipulate Objects. For example two classes are declared, Mouse and Lion, the two have a different kind of bite. Polymorphism means that a programmer can call a bite, without having to know from which Object does the bite comes, even though the user will be able to tell later whether the bite comes from a Lion or a Mouse, from the bite effect. Polymorphism

is given by Visual Basic (e.g. for windows) through a multiple interface called ActiveX (previously known as OLE or a short for Object Linking & Embedding), and not through inheritance. C^{++} developers are familiar with class hierarchies that describe inheritance. Inheritance is the manner in which objects are derived from similar objects, inheriting their behavior. Visual Basic OLE automation server object hierarchies, on the other hand, describe containment. Containment describes how an object can contain other objects. Multiple Interface facility is one superior advantage of Component Object Model (more commonly known now as DCOM or Distributed Component Object Model) over the conventional programming, because it enables a program to evolve, by including new functions, without damaging the existing programming Code, even for a multi user environment.

3. Examples & their relation to the advancement of Higher Education

3.1 Library Automation

This application software was designed to overcome inefficiencies in data access, especially to manage book collection and circulation. The approach of bar-coding, database search and management is, of course, applicable also for company asset tracking, shops and department stores' purchasing process. For this application, the database used in MCU was MS Access, and in UPH is MS SQL (still in progress). As can be seen in figure 1, there are four main Forms: Members management, cataloging, Book Collection management and Circulation. Each Button's enable property can be enabled or disabled, thus the application can be adapted easily for student's use (i.e. Button Catalogue only will be enabled) or for staff's use (all Buttons are enabled).

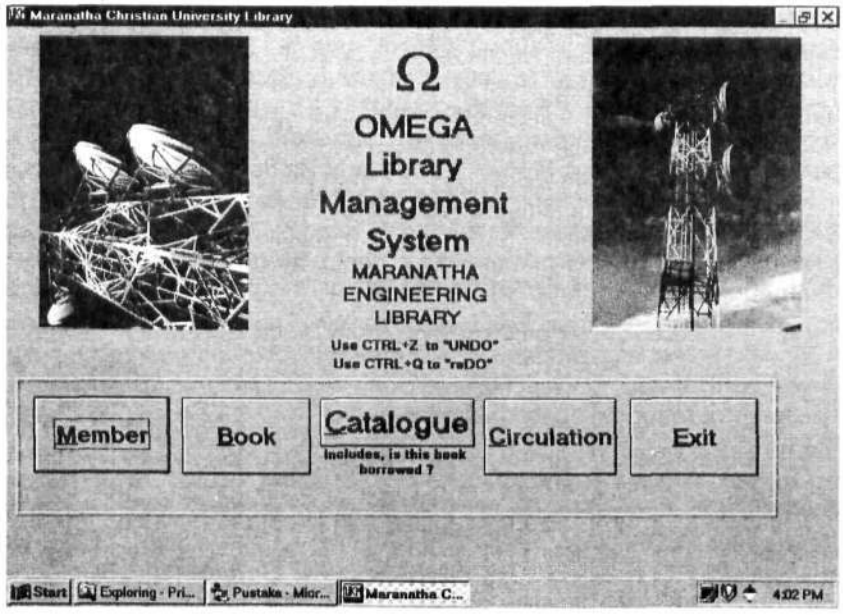

Figure 1, The Main Menu of Library Automation software

Next is the Circulation Form (Figure 2), which is called after the Circulation button in figure 1, is pressed (Note: for staff's set up, a password form must be filled in first, which is not shown here). In this Form, after the Library staff enters via a Keyboard or read through a Bar-code scanner, the student's ID number (or membership number), the cursor will be moved to the Book Bar-code data field (e.g. number *007004488, in Figure 2, where 007 means the engineering Dept. and 4488 is the book number), and the student's relevant data will be shown. Then, when the Bar-code is read from the Bar-code label on the book, the appropriate book data will appear, and the staff can either press Borrow (for borrowing case) or Return (in this case Borrow button will be disabled, because the borrowing date is already filled in). The Borrow and the Return dates are entered automatically by the software, utilizing the server's system clock. In this case the software, the server and the connected workstations were already tested for year 2000 compliance.

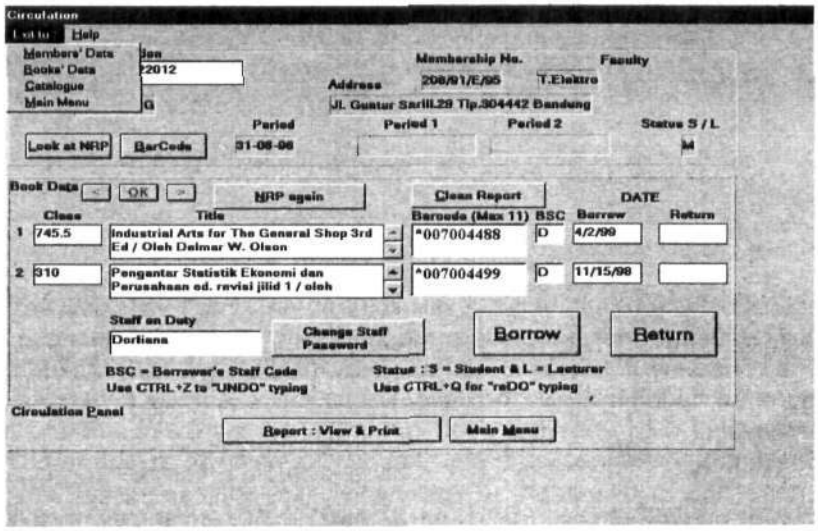

Figure 2. Circulation Form

Besides Circulation, Book Collection also needs management, Figure 3, shows this facility. Besides, the common facilities such us Class, Sub Class, ISBN, Title etc. Another facility is the ability to print Bar-codes out on plain A4 papers as well as on special sizes using ordinary printer.

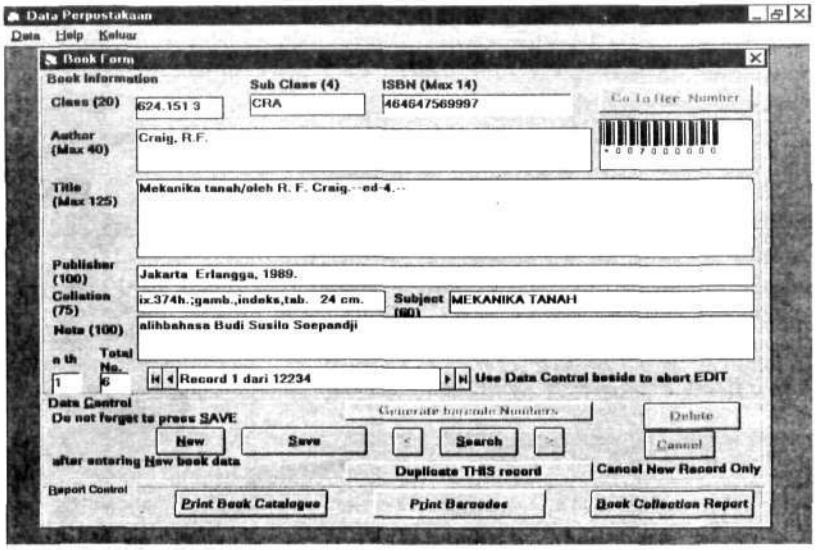

Figure 3. Book Management Form

3.2 Automated Quiz (AutoQuiz)

AutoQuiz is designed to help students in their studies – giving them quick feed back to their current knowledge of the subject of their interest. It can also help lecturers in quickly assessing their quality of teaching and students' current perception of the material being taught. More over, since students' result will be entered in to the server's database automatically – it will cut down the lecturer's time spent for examining and scoring each student's work. There are a number of important factors considered in automating a quiz :

- More than one subject should be available for quick assessments, in Figure 4 for example, there are two subjects available. Also, participation from other lecturers, to help to make a bank (e.g. of 50) of multiple choice questions and answers, is beneficial) – to socialize the use of this tool, a series of presentations should be conducted.
- Student's identification still needs to be confirmed, even though each quiz participant must fill in his or her name and student's ID number (Figure 5).
- Questions (in this case 20 out of 50 questions) and answers' position should be randomized to avoid cheating by looking across to other student's answers, as can be seen in Figure 6. (e.g. student 1 say has question 19, with the correct answer at "A", although has the same question at number 5, to student 2, sitting next to him/her, has the correct answer at "D")
- The Automated Quiz should be relatively easy to use – mouse control and short cut – keys, should be available.
- In Figure 7, student's score and time spent are available immediately after answering the last question (the student ID number is displayed again, so that the supervisor or lecturer can note down the result and re-check the student's identification card (should be with a photo). This information also goes automatically into the database. The lecturer can see the results immediately on his computer display, while the quiz is still in progress. After the quiz, the table of results can be printed or saved easily.

- A form to create new, modify, delete, or add multiple choice Question and answer should be available for lecturers, e.g. Figure 8, in another version of AutoQuiz program – restricted in its use by password system in AutoQuiz and user level permission in Windows NT.
- Essay type questions can be entered, and availability of input field for program path in file server (useful for "how to program test")
- The language and database software chosen should be able to display pictures and diagrams, in order to make more complex multiple-choice questions (e.g. A diagram of a computer shown in figure 6).
- The server's power-line should be backed-up, by using an Uninterruptible Power Supply (UPS).
- The database system should be reliable i.e. can be repaired and compacted automatically, by the AutoQuiz when it is necessary (to solve data corruption and server hard disk-space wastage), and should be secure for the networking environment (applying passwords for data base and users – with varying privileges).

The above factors had been considered in the AutoQuiz implemented, as shown in figure 4 to 8. The database used currently is MS Access. Actual Automated Quiz was conducted for the subjects such as: Telecommunications, and Industrial Electronics in the period of February up to May 1999, at Maranatha Christian University – and from October 2001 till now, for the subjects such as: Human Computer Interaction, Computer Security as well as, for Laboratory Assistants Selection test, at Universitas Pelita Harapan Computer Laboratories, with promising results i.e. benefiting both the students and lecturer.

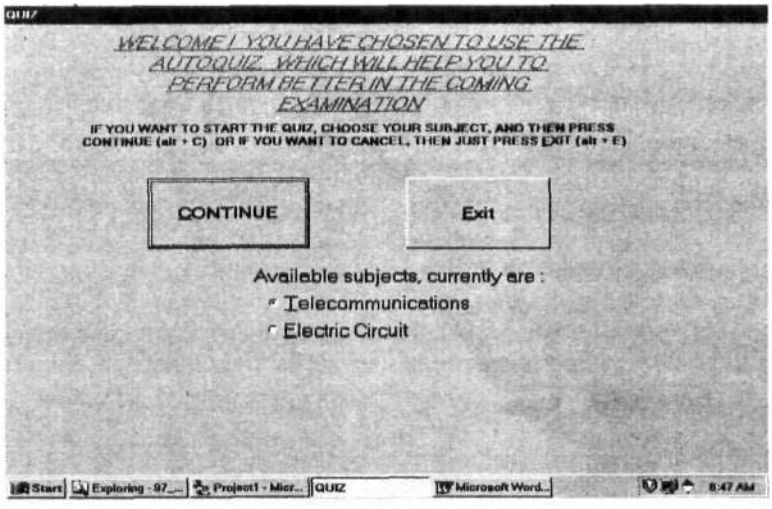
QUIZ
WELCOME ! YOU HAVE CHOSEN TO USE THE AUTOQUIZ WHICH WILL HELP YOU TO PERFORM BETTER IN THE COMING EXAMINATION
IF YOU WANT TO START THE QUIZ, CHOOSE YOUR SUBJECT, AND THEN PRESS CONTINUE (alt + C) OR IF YOU WANT TO CANCEL, THEN JUST PRESS EXIT (alt + E)
CONTINUE
Exit
Available subjects, currently are :
Telecommunications
Electric Circuit
Start
Exploring - 97_...
Project1 - Micr...
QUIZ
Microsoft Word...
8:47 AM

Figure 4. Auto Quiz Main Menu

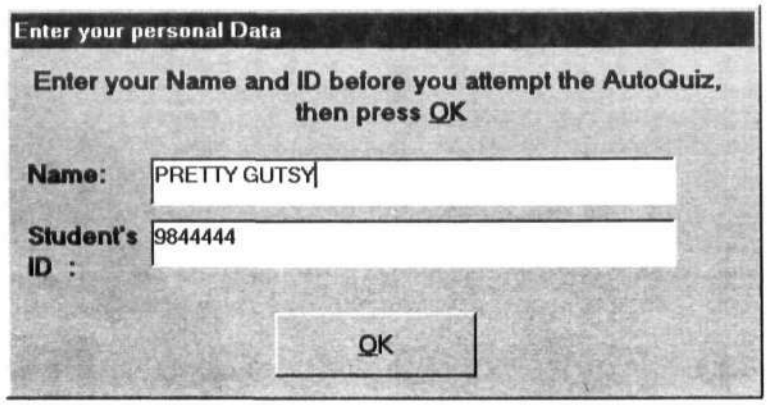

Figure 5. Entering Personal data before the Quiz is started

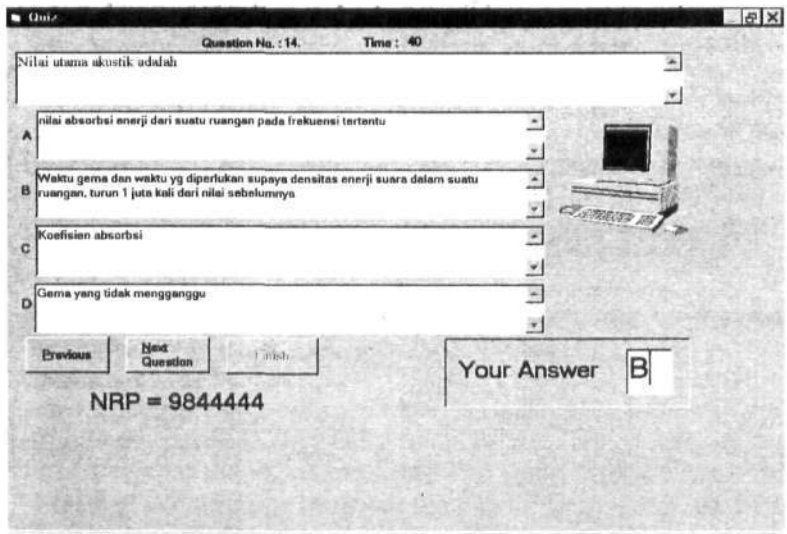

Figure 6. Randomized Multiple Choice Questions for the student

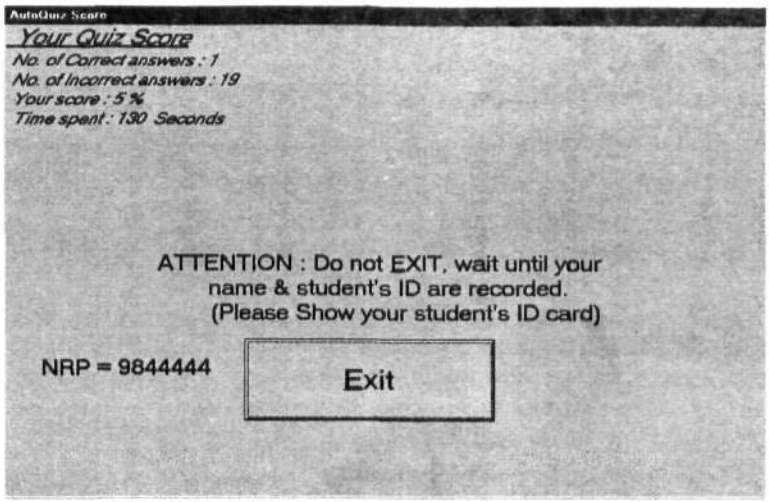
AutoQuiz Score
Your Quiz Score
No. of Correct answers : 1
No. of Incorrect answers : 19
Your score : 5 %
Time spent : 130 Seconds
ATTENTION : Do not EXIT, wait until your
name & student's ID are recorded.
(Please Show your student's ID card)
NRP = 9844444
Exit

Figure 7. Student's score is available immediately after answering the last question

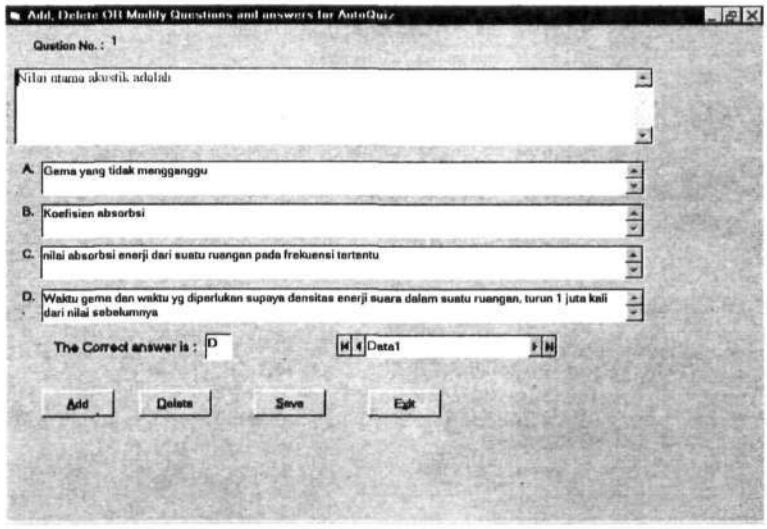
Add, Delete OR Modify Questions and answers for AutoQuiz
Question No. : 1
Nilai utama akustik adalah
A. Gema yang tidak mengganggu
B. Koefisien absorbsi
C. nilai absorbsi energi dari suatu ruangan pada frekuensi tertentu
D. Waktu gema dan waktu yg diperlukan supaya densitas energi suara dalam suatu ruangan, turun 1 juta kali dari nilai sebelumnya
The Correct answer is : D
Data1
Add
Delete
Save
Exit

Figure 8. Question and answer form for the lecturer

3.3 Simulation Program

Software Simulations can help students understand better a particular subject of their interest. It can be a subject with laboratory or without laboratory work. Further more, students can "see" expensive equipment at work, without the necessity for the university to have to purchase the actual equipment. Unfortunately, even this kind of application software, is not easily available in the market and often are relatively expensive to buy. The following is two examples of simulation program, just to illustrate that OOP can be used to make customized simulation software.

Fisheries sonar is a very useful equipment, especially for a maritime country like Indonesia. However, this equipment is usually imported. Figure 9 shows how an actual Fisheries Sonar can work. As a start, the location of school of fish (up to 9 schools of fish can be generated randomly) is generated randomly by the program and as the school of fish moves, the angle, its radial-distance, as well as speed, from the sonar are shown in a table. It can be adapted and calibrated, in the future, to receive the angle and distance data from an actual sonar equipment (e.g. through a serial port), which scans radially (of a circle area up to 16 km or 16,000 meter in diameter).

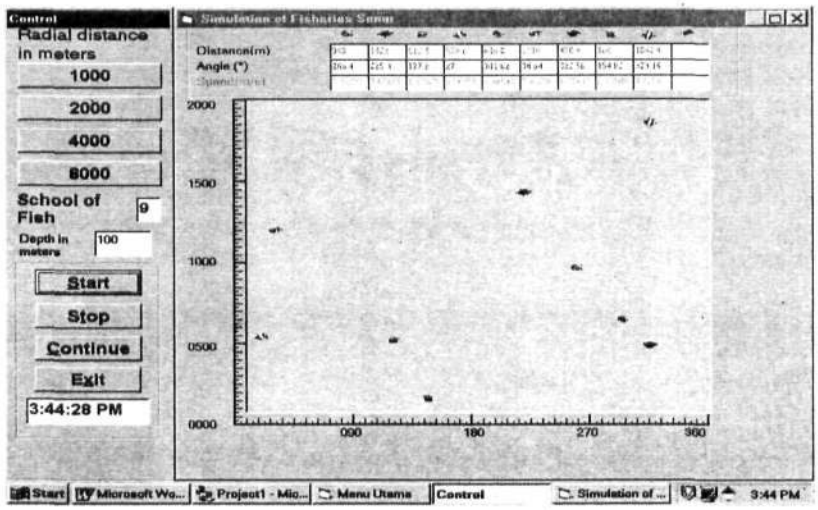

Figure 9. Simulation of Fisheries Sonar

The second example, Figure 10, shows that a student can find out the minimum required distance for a moving vehicle to come to a full stop, after for example, an initial speed of 100 km/h, and a strong application of the break pedal which has caused a negative acceleration of 10m/sec^2, applied when the vehicle's rear still touching the long pole / marker on the left hand side. For further investigation, the road angle (at a slope / an inclination, instead of, just horizontal), vehicle's weight & its centre of gravity, as well as, road type (smooth or rough) can be varied.

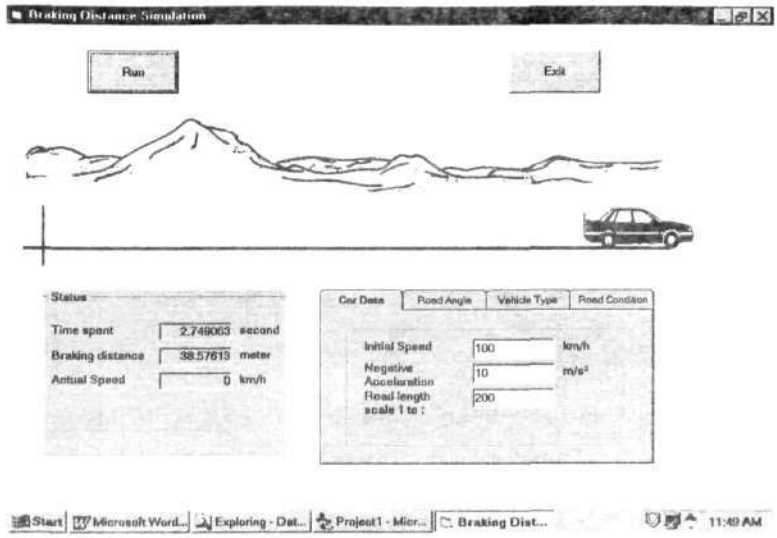

Figure 10. Vehicle's Braking Distance Simulation

4. Summary

Component programming enables a programmer to add new functions, re-use existing group of programming codes (e.g. under a Form or Module), and even generate multi user customized-software with relative ease. However, because it is inherently event driven a program must be planned well before the actual write-up. This is a must, especially, if a team of programmers writes the program-code. If customized software can be written by the institutions' programmers according to their needs, it can contribute in improving the quality of local higher education system.

Acknowledgement

The author would like to express much gratitude to Maranatha Christian University & its foundation, the Chancellor (Dr Daniel Wibowo MSc.), Vice Chancellor (Mrs. A. Supartono, Ir., MSc.), for their moral and financial support. Not forgetting, also, the Chancellor's secretary plus other staffs (too many to mention), who had been helpful in one way or another, so that this paper can be presented in this seminar*. In addition, my thanks goes to many final year students – (a few have already become alumni, like, Andreeyanto V., Yoswandi H., Paulus E.P., and Budyanto), who had helped me in the initial development of the electronic library, data entry etc., as well as, Gunawan E., & Wiharja J.W, who had helped me in the initial development of Automatic Quiz system (single user) and last but not least, Handini V., & Tedja S., who had participated in the development of simulation program.

References

1. Firdaus C., Tjong W.S., Sistem Pengaman Network pada Paket Lunak Berbasis Windows 95 / NT *(Network security system for Windows 95 / NT based software)* , Kerja Praktek dengan Pembimbing : B. Hardjono, 1998.

2. Goren C., Scmelzer J., Smith J., *Visual Basic 4*, Enterprise Development, QUE, 1996, pp. 165, 304
3. Hardjono B., Otomatisasi akses data dengan menggunakan basis data dan kode batang *(Automatic data access using database and bar-code system)*, 1998 (majalah ilmiah UKM / *University's Research magazine*)
4. Hardjono B., Otomatisasi Akses Informasi dengan menggunakan perangkat lunak untuk sistem Kode Batang (Bar Code System) dan pemanfaatan perangkat kerasnya *(Automatic information access utilizing custom-made bar-coding software and its hardware)*, 1997 (Laporan Penelitian / *A research report*)
5. Hardjono B., *S.A.S Tool, Siemens' Automated Survey Tool*, a Windows based application software made for Siemens Telecommunication Project Office, User's manual and software tool copy-right : B. Hardjono and Siemens' TPO, Bandung, Indonesia, 1995.
6. McKelvy M., Martinsen RR, Web J., Riselman B., *Using Visual Basic 5*, Special Edition, QUE, 1997; chapter title: Applying Computer Science Techniques in Visual Basic, sub chapter: Advanced Code Techniques.
7. Microsoft Corporation, *Programmer's Guide MS Visual Basic Version 5, Programming System for Windows 95, dan Windows NT*, p.5 - 337-, Microsoft Corporation, 1997.
8. Microsoft Corporation, *Visual Basic Guide to Building Client/Server, MS Visual Basic 5, Programming System for Windows 95 and Windows NT*, MS Corp., 1997, pp. 101
9. Perry P., Teach Yourself Visual Basic 3 in 21 Days, Sams Publishing, 1994
10. Salemi J., Client/Server Databases, Ziff-Davis Press, 1993